SECOND EDITION

Computer Graphics

PRINCIPLES AND PRACTICE

James D. Foley
The George Washington University

Andries van Dam
Brown University

Steven K. Feiner
Columbia University

John F. Hughes
Brown University

ADDISON-WESLEY PUBLISHING COMPANY
Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam • Bonn
Sydney • Singapore • Tokyo • Madrid • San Juan

BA

Sponsoring Editor: Keith Wollman
Production Supervisor: Bette J. Aaronson
Copy Editor: Lyn Dupré
Text Designer: Herb Caswell
Technical Art Consultant: Joseph K. Vetere
Illustrators: C&C Associates
Cover Designer: Marshall Henrichs
Manufacturing Manager: Roy Logan

This book is in the **Addison-Wesley Systems Programming Series**
Consulting editors: IBM Editorial Board

Library of Congress Cataloging-in-Publication Data
Computer graphics: principles and practice/James D. Foley . . . [et al.].—2nd ed.
p. cm.
Includes bibliographical references.
ISBN 0-201-12110-7
1. Computer graphics. I. Foley, James D., 1942–.
T385.C587 1990
006.6—dc20 89-35281
CIP

Cover: "Dutch Interior," after Vermeer, by J. Wallace, M. Cohen, and D. Greenberg, Cornell University (Copyright © 1987 Cornell University, Program of Computer Graphics.)

ABCDEFGHIJ-DO-943210

To Marylou, Heather, Jenn, my parents, and my teachers
Jim

To Debbie, my father, my mother in memoriam, and my children Elisa, Lori, and Katrin
Andy

To Jenni, my parents, and my teachers
Steve

To my family, my teacher Rob Kirby, and my father in memoriam
John

And to all of our students.

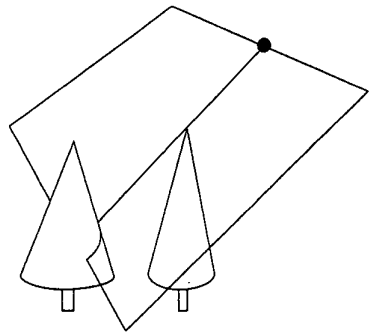

Fig. 20.19 The plane from an adjacent tree, which determines the shadowing of a tree.

tree is in shadow from another tree, the specular and diffuse components should not be added. This was implemented by determining planes from nearby trees to the tree under consideration; the plane contained the top of the nearby tree and the light source, and had the largest possible y component in its normal, as shown in Fig. 20.19. Particles above this plane were lit with all three components, whereas those below were given (probabilistically) less and less diffuse and specular light as the distance from the plane increased.

Even with these simplified lighting computations, visible surfaces still had to be determined. The trees in the scene were sorted back to front, and were rendered in that order. Trees were rendered with a bucket-sort type of z-buffer. Each tree's depth extent was divided into a great many buckets; every particle that was generated was inserted into the bucket for its depth in the tree. When all particles had been generated, they were rendered in back-to-front bucket order. Each particle was drawn as a small circle or short line segment (antialiased). After each tree was rendered, the information about the tree was discarded. The result of this ordering is that a branch of a nearby tree may obscure a branch of one slightly farther away, even though the second branch lies in front of the first, since the first branch is part of a tree that is rendered (entirely) after the second. In scenes with sufficient complexity, this sorting error seems not to be a problem.

Still, this difficulty in rendering the scene does highlight a drawback of particle systems in general: The modeler gets considerable power, but special-purpose rendering techniques may need to be developed for each new application.

20.6 VOLUME RENDERING

Volume rendering is used to show the characteristics of the interior of a solid region in a 2D image. In a typical example, the solid is a machined part that has been heated, and the temperature has been computed at each point of the interior through some physical or mathematical means. It is now of interest to display this temperature visually. This is not, strictly speaking, a modeling issue, as the shape of the part and the characteristics to be displayed are both available a priori. But the conversion of these data to information in a pixel map is a form of modeling; namely, the modeling of the transformation from 3D to 2D. In another example, the density of human or animal tissue may have been computed at each point of a 3D grid through computed tomography (CT). The display of this information should indicate the boundaries of the various types of tissue (as indicated by density changes). The surfaces defining these boundaries must be inferred from the sample data in order to render the solid.

A number associated with each point in a volume is called the *value* at that point. The collection of all these values is called a *scalar field* on the volume. The set of all points in the volume with a given scalar value is called a *level surface* (if the scalar field is sufficiently continuous, this set of points actually does form a surface). *Volume rendering* is the process of displaying scalar fields. It is important to realize that the data being displayed may not be ideal. If the data have been sampled at the points of a regular grid, the scalar field they represent may contain frequencies higher than the Nyquist frequency for the sampling (see Chapter 14). In tomography, for example, the transition from flesh to bone is very abrupt, and hence contains very high frequencies, but the sampling rate is likely to be too low to represent this change accurately. Also, the data that describe the interior of a solid may be clustered in some irregular pattern, as might arise in geographic data taken from core samples, where it may be impossible to sample uniformly.

Several approaches to volume rendering have been developed. They can be divided into two categories: those that compute level surfaces and those that display integrals of density along rays. The two can be combined by assigning density only to certain level surfaces and then ray tracing the result (which amounts to creating a different volume to be displayed). If animation is available, a third category of display is possible: a series of 2D slices of the data is computed and displayed sequentially, using color or brightness to indicate the scalar value at each point of the slices. If interactive control of the slice direction and level is provided, this approach can give an excellent sense of the interior structure of the scalar field.

Nonetheless, it is sometimes useful to view data in the aggregate, rather than by slices. A first step in this direction was provided by the *marching-cubes* algorithm. In this algorithm, scalar values are assumed to be given at each point of a lattice in 3-space. A particular level surface can be approximated by determining all intersections of the level surface with edges of a lattice.[6] We look for pairs of adjacent lattice points whose field values surround the desired value (i.e., the value of one vertex is greater than the chosen level, the value of the other is less). The location of an intersection of the level surface with the edge is then estimated by linear interpolation.

Each cube in the lattice now has some number of edges marked with intersection points. The arrangement of the intersection points on the edges can be classified into 256 cases (each of eight vertices of each cube in the lattice is either above or below the target value, giving $2^8 = 256$ possible arrangements). For each case, a choice is made of how to fill in the surface within the cube. Figure 20.20 shows two such cases.

[6] A *lattice* is an array of points and lines in space, much like a children's jungle gym. The points of the lattice are evenly spaced in the x, y, and z directions, and they are joined by line segments parallel to the coordinate axes. The set of all points with integer coordinates and of all axis-parallel line segments joining them constitutes an example, called the *integer lattice*.

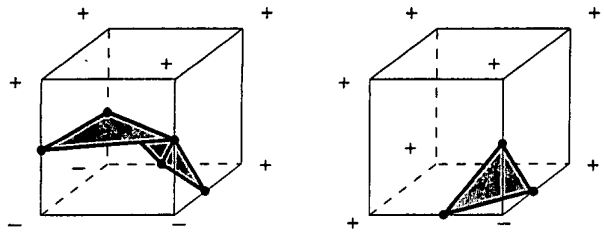

Fig. 20.20 Two possible arrangements of intersections of a level surface with a cube in the integer lattice, with choices of how to fill in a surface for each.

The collection of all the surface pieces just defined constitutes a surface. This surface can be assigned (at each subpolygon) a normal vector to be used for shading in the following manner. At each vertex of the cube, a numerical estimate of the gradient of the scalar field is made. These values are interpolated to estimate the gradient vector at some point of the subsurface. Since the gradient of a scalar field always lies in the direction of the normal vector to the level surface, this interpolated value provides a good estimate for the normal vector. (The special case of zero must be handled separately.)

The resulting level surface can be rendered with conventional techniques. This strategy can be of use in medical imaging to show the shape of the boundary between different types of tissue. Unfortunately, it computes only one shape at a time, and the relative positions of different layers are difficult to see.

Upson and Keeler [UPSO88] also assume that the scalar field varies linearly between sample points, and they present two methods for its display. In both, the user first creates four functions, R, G, B, and O, where O is *opacity*. The arguments of these functions are values of the scalar field; we therefore assume that the scalar field has been normalized to have values between 0 and 1. The choices of the R, G, B, and O functions drastically affect the resulting image. If the functions are chosen to have tight peaks at particular values of the scalar field, the level surfaces for those values are highlighted. If the functions are chosen to vary smoothly over the field values, then color can be used to indicate field value (see Fig. 20.21). Thus, in effect, we obtain sophisticated color-map pseudocoloring.

The interpolation of the scalar field over each cube in the lattice of sample points is a linear equation in each variable, and hence is trilinear in 3-space (i.e., of the form $S(x, y, z) = A + Bx + Cy + Dz + Exy + Fxz + Gyz + Hxyz$). If we parameterize a ray in the form $(x, y, z) = (a, b, c) + t(u, v, w)$ as in ray tracing, then the value of S at points of the ray is a cubic function of t.

The ability to compute this cubic rapidly forms the basis for Upson and Keeler's first rendering method, based on a ray-tracing mechanism for volume data developed in [KAJI84]. For each ray from the eyepoint through an image pixel, the R, G, B, and O values are accumulated for the ray as it passes through the volume data. This accumulation stops when the opacity reaches a value of 1 or the ray exits the volume, whichever happens first. Actually, far more is accumulated: the scalar field, shading function, opacity, and depth cueing are all computed at each of several steps within each pixel volume so as to integrate the cubic interpolant accurately.

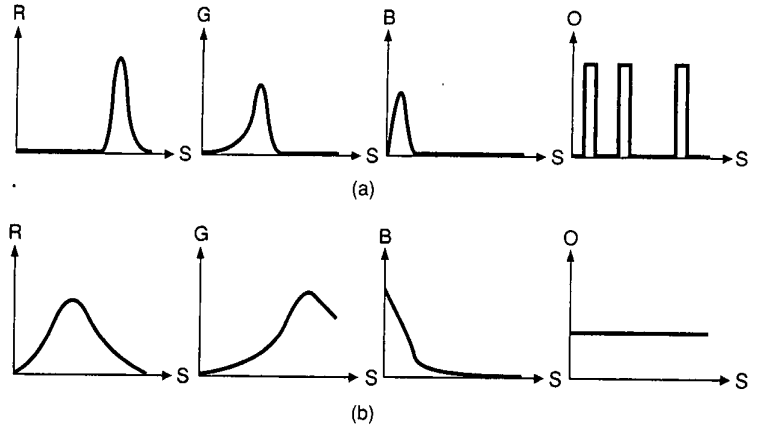

Fig. 20.21 Two different choices for the shapes of the R,G,B, and O functions. In (a), certain level surfaces of the scalar field are highlighted in red, green, and blue. In (b), the color will change gradually as a function of the scalar field.

Upson and Keeler's second rendering method uses the same basic notion of integration along rays, but accumulates values in pixels by processing the cubes in the lattice of values in front-to-back order (which can be easily determined for any particular view orientation). The authors take great pains to ensure the computational efficiency of the process by using adaptive quadrature methods for the integrations and never solving a system of equations more than once at each point (when performing interpolations). It is important to observe, as they do, that this method "is designed as an analytic tool, not as a technique to synthesize realistic images" [UPSO88, p. 64].

Sabella takes a similar approach [SABE88]. He assigns a *density emitter* to each point in the volume to be rendered, to simulate light coming from translucent objects. The simulation models only part of the effects of light in such media; namely, the occlusion of parts deeper in the medium by those nearer the front. Sabella deliberately ignores shadowing and the variation in color due to differences in scattering at different wavelengths, asserting that they may actually detract from the perception of density variation. The density emitters are imagined to be tiny particles that both emit and scatter light. The density of such particles within each small region of space is given by the value of the scalar field there. The light reaching the eye along any ray is computed by summing up the emission from all the emitters along the ray, and then attenuating the light from each emitter by the probability that it is scattered during its travel to the eye. Sabella computes four numbers: M, the peak value of the scalar field along the ray; D, the distance at which that peak is encountered; I, the attenuated intensity just described; and C, the "center of gravity" of the density emitters along the ray. By mapping combinations of these numbers into various color scales (e.g., using hue–saturation–value, he maps M to hue, D to saturation, and I to value), he can highlight various characteristics of the scalar field. He further allows for "lighting" effects by giving a directionality to the particle emissions. Each particle's emissions are attenuated by a Lambert lighting model: Several light sources

are positioned around the volume to be rendered, and the emission from a particle at location (x, y, z) is determined by summing the dot products of the gradient of the scalar field and the lighting directions, and multiplying the result by the density at the point. The result is that surfaces of high density look more like reflective surfaces, an effect that helps the eye to disambiguate the information presented.

Even further from the determination of surfaces is the approach taken by Dreben, Carpenter, and Hanrahan at Pixar [DREB88]. These researchers make several important assumptions about the scalar fields being rendered: the volume array of data representing the field is assumed to be sampled at about the Nyquist frequency of the field (or the field has been filtered to ensure this before sampling); the scalar field is modeled by a composition of one or more materials (e.g., bone, fat, and soft tissue) or the volume has several scalar fields attached to it, such as stress and strain in a material. For a multiple-material scalar field, they assume that the materials can be (at least statistically) differentiated by the scalar value at each point, or that information regarding the material composition of each volume element is provided in addition to the scalar field.

Given such information, they create several new scalar fields on the array of sample points: the *material percentage volumes* (they use the term *volume* to mean a scalar field on a volume). The value at a grid point in a material percentage volume is the percentage of one material present in the volume element (or *voxel*) surrounding that point. If multiple fields are specified in the original data, computing these material percentages may be simple. If only a single field is given, the material percentages may have to be estimated by Bayesian analysis.

After computing the material percentage volumes, the authors associate a color and opacity with each material; they then form composite colors and opacities by taking a linear combination of all the colors and opacities for each of the material percentage volumes. (Opacity here is used in the sense of the α channel described in Section 17.6, and the linear combinations are the same as the combinations described there. In particular, the colors are premultiplied by the opacity values before combination.) They further allow compositing with *matte volumes*, which are scalar fields on the volume with values between 0 and 1. By multiplying these matte volumes with the color/opacity volumes, they can obtain slices or portions of the original volumetric data. Making a smooth transition between 0 and 1 preserves the continuity of the data at the matte boundaries.

The lighting model used here is similar to that in the other two algorithms. A certain amount of light enters each voxel (the light from voxels behind the given voxel), and a different amount exits from it. The change in light can be affected by the translucence of the material in the voxel, or by "surfaces" or "particle scatterers" contained in the voxel that may both attenuate the incoming light and reflect light from external light sources. These effects are modeled by (1) requiring that light passing through a colored translucent voxel have the color of that voxel plus the incoming light multiplied by $(1 - \alpha)$ for that voxel (this is the **over** operation of the Feibush–Levoy–Cook compositing model in Section 17.6), and (2) determining surfaces and their reflectance and transmission properties.

The surface determination is not as precise as the ones described previously; each voxel is assigned a density that is a weighted sum of the densities of the component materials for the voxels (weighted by the material percentages). "Surfaces" are simply places where this

composite density changes rapidly. The *strength* of a surface is the magnitude of the gradient of the density, and the surface normal used in shading calculations is the direction vector of the gradient. To compute the surface shading, we divide each voxel into regions in front of, on, and behind the surface. The intensity of light leaving the voxel, I', is related to the intensity entering, I, by the rule $I' = (C_{\text{front}}$ **over** $(C_{\text{surface}}$ **over** $(C_{\text{back}}$ **over** $I)))$. The three terms associated with the voxel can be precomputed and mixed because the **over** operator is associative. The surface color is computed by a Cook–Torrance–style model to give both specular and diffuse components; these values are weighted by the strength of the surface so that no reflective lighting appears in homogeneous solids. The colors of the front and back are computed by estimating from which material they came and by using colors from those materials.

The results are excellent. Color Plate IV.1 shows the process as applied to data from a CT scan of a human body. The process is expensive, however. Multiple volumes (i.e. multiple scalar fields) are created in the course of generating the image, and the memory requirements are vast. Also, the assumption that the fields are sampled at or above the Nyquist frequency may not be practical in all cases: sometimes, the data are given, and we wish to see the results even with some aliasing. Finally, the assumption that the data are from a heterogeneous mixture of materials is not always valid, so the applications of the method are limited.

20.7 PHYSICALLY BASED MODELING

The behavior and form of many objects are determined by the objects' gross physical properties (as contrasted with biological systems, whose behavior may be determined by the systems' chemical and microphysical properties). For example, how a cloth drapes over objects is determined by the surface friction, the weave, and the internal stresses and strains generated by forces from the objects. A chain suspended between two poles hangs in an arc determined by the force of gravity and the forces between adjacent links that keep the links from separating. *Physically based modeling* uses such properties to determine the shape of objects (and even their motions in some cases). Current work on this subject is collected in [BARR89].

Most of this modeling uses mathematics well beyond the scope of this book, but we can give the general notions of the techniques. It is in this sort of modeling that the distinction between graphics and other sciences is most blurred. The computations that produce a tear in a model of a thin cloth when it is dropped over an obstruction are purely in the domain of solid mechanics. But such computations would not be done unless the results could be displayed in some fashion, so the motivation for physical research is now being provided by the ability (or desire) to visualize results. At the same time, the wish to generate more realistic graphics models drives research in the physical modeling process. In this section, we discuss a few of the more impressive examples. The next section describes models of natural phenomena that are less directly based on scientific principles and may contain some (or many) compromises in order to produce attractive results. There is a continuous variation between scientific foundations and ad hoc approaches, and the dividing line is not at all clear.

Volume Illustration: Non-Photorealistic Rendering of Volume Models

David Ebert Penny Rheingans
Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore MD 21250
[ebert | rheingan]@cs.umbc.edu

Abstract

Accurately and automatically conveying the structure of a volume model is a problem not fully solved by existing volume rendering approaches. Physics-based volume rendering approaches create images which may match the appearance of translucent materials in nature, but may not embody important structural details. Transfer function approaches allow flexible design of the volume appearance, but generally require substantial hand tuning for each new data set in order to be effective. We introduce the volume illustration approach, combining the familiarity of a physics-based illumination model with the ability to enhance important features using non-photorealistic rendering techniques. Since features to be enhanced are defined on the basis of local volume characteristics rather than volume sample value, the application of volume illustration techniques requires less manual tuning than the design of a good transfer function. Volume illustration provides a flexible unified framework for enhancing structural perception of volume models through the amplification of features and the addition of illumination effects.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – color, shading, and texture; I.3.8 [Computer Graphics]: Applications.
Keywords: Volume rendering, non-photorealistic rendering, illustration, lighting models, shading, visualization.

1 Introduction

For volume models, the key advantage of direct volume rendering over surface rendering approaches is the potential to show the structure of the value distribution throughout the volume, rather than just at selected boundary surfaces of variable value (by isosurface) or coordinate value (by cutting plane). The contribution of each volume sample to the final image is explicitly computed and included. The key challenge of direct volume rendering is to convey that value distribution clearly and accurately. In particular, showing each volume sample with full opacity and clarity is impossible if volume samples in the rear of the volume are not to be completely obscured.

Traditionally, volume rendering has employed one of two approaches. The first attempts a physically accurate simulation of a process such as the illumination and attenuation of light in a gaseous volume or the attenuation of X-rays through tissue [Kajiya84, Drebin88]. This approach produces the most realistic and familiar views of a volume data set, at least for data that has an appropriate physical meaning. The second approach is only loosely based on the physical behavior of light through a volume, using instead an arbitrary transfer function specifying the appearance of a volume sample based on its value and an accumulation process that is not necessarily based on any actual accumulation mechanism [Levoy90]. This approach allows the designer to create a wider range of appearances for the volume in the visualization, but sacrifices the familiarity and ease of interpretation of the more physics-based approach.

We propose a new approach to volume rendering: the augmentation of a physics-based rendering process with non-photorealistic rendering (NPR) techniques [Winkenbach94, Salisbury94] to enhance the expressiveness of the visualization. NPR draws inspiration from such fields as art and technical illustration to develop automatic methods to synthesize images with an illustrated look from geometric surface models. Non-photorealistic rendering research has effectively addressed both the illustration of surface shape and the visualization of 2D data, but has virtually ignored the rendering of volume models. We describe a set of NPR techniques specifically for the visualization of volume data, including both the adaptation of existing NPR techniques to volume rendering and the development of new techniques specifically suited for volume models. We call this approach *volume illustration*.

The volume illustration approach combines the benefits of the two traditional volume rendering approaches in a flexible and parameterized manner. It provides the ease of interpretation resulting from familiar physics-based illumination and accumulation processes with the flexibility of the transfer function approach. In addition, volume illustration provides flexibility beyond that of the traditional transfer function, including the capabilities of local and global distribution analysis, and light and view direction specific effects. Therefore, volume illustration techniques can be used to create visualizations of volume data that are more effective at conveying the structure within the volume than either of the traditional approaches. As the name suggests, volume illustration is intended primarily for illustration or presentation situations, such as figures in textbooks, scientific articles, and educational video.

2 Related Work

Traditional volume rendering spans a spectrum from the accurate to the ad hoc. Kajiya's original work on volume ray tracing for generating images of clouds [Kajiya84] incorporated a physics-based illumination and atmospheric attenuation model. This work in realistic volume rendering techniques has been extended by numerous researchers [Nishita87, Ebert90, Krueger91, Williams92, Max95, Nishita98]. In contrast, traditional volume rendering has relied on the use of transfer functions to produce artificial views of the data to highlight regions of interest [Drebin88]. These transfer functions, however, require in-depth knowledge of the data and need to be adjusted for each data set.

The design of effective transfer functions is still an active research area [Fang98, KindImann98, Fujishiro99]. While transfer functions can be effective at bringing out the structure in the value distribution of a volume, they are limited by their dependence on voxel value as the sole transfer function domain.

In contrast, there has been extensive research for illustrating surface shape using non-photorealistic rendering techniques. Adopting a technique found in painting, Gooch et al. developed a tone-based illumination model that determined hue, as well as intensity, from the orientation of a surface element to a light source [Gooch98]. The extraction and rendering of silhouettes and other expressive lines has been addressed by several researchers [Saito90, Salisbury94, Gooch99, Interrante95]. Expressive textures have been applied to surfaces to convey surface shape [Rheingans96, Salisbury97, Interrante97].

A few researchers have applied NPR techniques to the display of data. Laidlaw used concepts from painting to create visualizations of 2D data, using brushstroke-like elements to convey information [Laidlaw98] and a painterly process to compose complex visualizations [Kirby99]. Treavett has developed techniques for pen-and-ink illustrations of surfaces within volumes [Treavett00]. Interrante applied principles from technical illustration to convey depth relationships with halos around foreground features in flow data [Interrante98]. Saito converted 3D scalar fields into a sampled point representation and visualized selected points with a simple primitive, creating an NPR look [Saito94]. With the exceptions of the work of Saito and Interrante, the use of NPR techniques has been confined to surface rendering.

3 Approach

We have developed a collection of volume illustration techniques that adapt and extend NPR techniques to volume objects. Most traditional volume enhancement has relied on functions of the volume sample values (e.g., opacity transfer functions), although some techniques have also used the volume gradient (e.g., [Levoy90]). In contrast, our volume illustration techniques are fully incorporated into the volume rendering process, utilizing viewing information, lighting information, and additional volumetric properties to provide a powerful, easily extensible framework for volumetric enhancement. Comparing Diagram 1, the traditional volume rendering system, and Diagram 2, our volume illustration rendering system, demonstrates the difference in our approach to volume enhancement. By incorporating the enhancement of the volume sample's color, illumination, and opacity into the rendering system, we can implement a wide range of enhancement techniques. The properties that can be incorporated into the volume illustration procedures include the following:

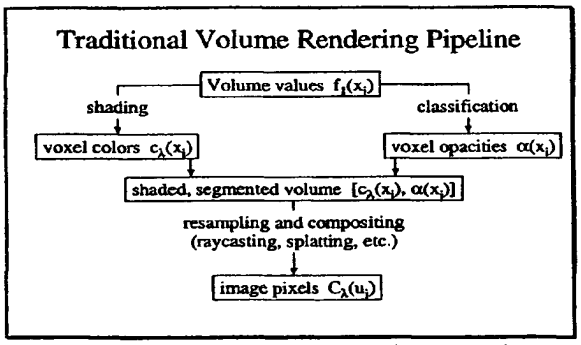

Diagram 1. Traditional Volume Rendering Pipeline.

- Volume sample location and value
- Local volumetric properties, such as gradient and minimal change direction
- View direction
- Light information

The view direction and light information allows global orientation information to be used in enhancing local volumetric features. Combining this rendering information with user selected parameters provides a powerful framework for volumetric enhancement and modification for artistic effects.

Volumetric illustration differs from surface-based NPR in several important ways. In NPR, the surfaces (features) are well defined, whereas with volumes, feature areas within the volume must be determined through analysis of local volumetric properties. The volumetric features vary continuously throughout three-dimensional space and are not as well defined as surface features. Once these volumetric feature volumes are identified, user selected parametric properties can be used to enhance and illustrate them.

We begin with a volume renderer that implements physics-based illumination of gaseous phenomena. The opacity transfer function that we are using is the following simple power function:

$$o_v = (k_{os} v_i)^{k_{oe}}$$

where v_i is the volume sample value and k_{os} is the scalar controlling maximum opacity. Exponent k_{oe} values less than 1 soften volume differences and values greater than 1 increase the contrast within the volume.

Figure 1 shows gaseous illumination of an abdominal CT volume of 256×256×128 voxels. In this image, as in others of this dataset, the scene is illuminated by a single light above the volume and slightly toward the viewer. The structure of tissues and organs is difficult to understand. In Figure 2, a transfer function has been used to assign voxel colors which mimic those found in actual tissue. The volume is illuminated as before. Organization of tissues into organs is clear, but the interiors of structures are still unclear. We chose to base our examples on an atmospheric illumination model, but the same approach can be easily applied to a base renderer using Phong illumination and linear accumulation.

In the following two sections, we describe our current collection of volume illustration techniques. These techniques can be applied in almost arbitrary amounts and combinations, becoming a flexible toolkit for the production of expressive images of volume models. The volume illustration techniques we

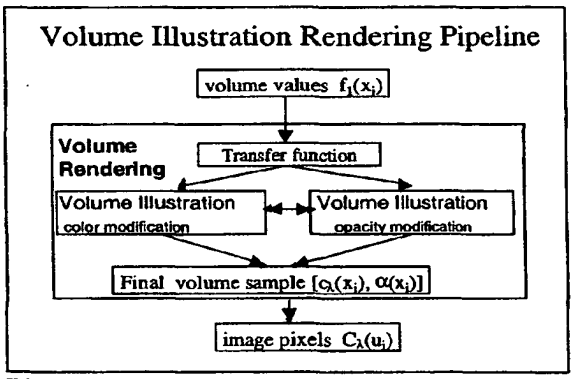

Diagram 2. Volume Illustration Rendering Pipeline.

propose are of two basic types: feature enhancement, and depth and orientation cues.

4 Feature Enhancement

In a surface model, the essential feature is the surface itself. The surface is explicitly and discretely defined by a surface model, making "surfaceness" a boolean quality. Many other features, such as silhouettes or regions of high curvature, are simply interesting parts of the surface. Such features can be identified by analysis of regions of the surface.

In a volume model, there are no such discretely defined features. Volume characteristics and the features that they indicate exist continuously throughout the volume. However, the boundaries between regions are still one feature of interest. The local gradient magnitude at a volume sample can be used to indicate the degree to which the sample is a boundary between disparate regions. The direction of the gradient is analogous to the surface normal. Regions of high gradient are similar to surfaces, but now "surfaceness" is a continuous, volumetric quality, rather than a boolean quality. We have developed several volume illustration techniques for the enhancement of volume features based on volume gradient information.

4.1 Boundary Enhancement

Levoy [Levoy90] introduced gradient-based shading and opacity enhancement to volume rendering. In his approach, the opacity of each voxel was scaled by the voxel's gradient magnitude to emphasize the boundaries between data (e.g., tissue) of different densities and make areas of constant density transparent (e.g., organ interiors). We have adapted this idea to allow the user to selectively enhance the density of each volume sample by a function of the gradient. Assume a volume data set containing a precomputed set of sample points. The value at a location P_i is a scalar given by:

$$v_i = f(P_i) = f(x_i, y_i, z_i)$$

We can also calculate the value gradient $\nabla_f(P_i)$ at that location. In many operations we will want that gradient to be normalized. We use ∇_{fn} to indicate the normalized value gradient.

Before enhancement, voxel values are optionally mapped through a standard transfer function which yields color value c_v and opacity o_v for the voxel. If no transfer function is used, these values can be set to constants for the whole volume. The inclusion of a transfer function allows artistic enhancements to supplement, rather than replace, existing volume rendering mechanisms.

We can define a boundary-enhanced opacity for a voxel by combining a fraction of the voxel's original opacity with an enhancement based on the local boundary strength, as indicated by the voxel gradient magnitude. The gradient-based opacity of the volume sample becomes:

$$o_g = o_v\left(k_{gc} + k_{gs}\left(\|\nabla_f\|\right)^{k_{ge}}\right)$$

where o_v is original opacity and ∇_f is the value gradient of the volume at the sample under consideration. This equation allows the user to select a range of effects from no gradient enhancement (k_{gc}=1, k_{gs}=0) to full gradient enhancement (k_{gs} >=1) to only showing areas with large gradients (k_{gc}=0), as in traditional volume rendering. The use of the power function with exponent k_{ge} allows the user to adjust the slope of the opacity curve to best highlight the dataset.

Figure 3 shows the effect of boundary enhancement in the medical volume. The edges of the lungs and pulmonary vasculature can be seen much more clearly than before, as well as some of the internal structure of the kidney. Parameter values used in Figure 3 are $k_{gc} = 0.7$, $k_{gs} = 10$, $k_{ge} = 2.0$.

4.2 Oriented Feature Enhancement: Silhouettes, Fading, and Sketch Lines

Surface orientation is an important visual cue that has been successfully conveyed by artists for centuries through numerous techniques, including silhouette lines and orientation-determined saturation effects. Silhouette lines are particularly important in the perception of surface shape, and have been utilized in surface illustration and surface visualization rendering [Salisbury94, Interrante95]. Similarly, silhouette volumes increase the perception of volumetric features.

In order to strengthen the cues provided by silhouette volumes, we increase the opacity of volume samples where the gradient nears perpendicular to the view direction, indicated by a dot product between gradient and view direction which nears zero. The silhouette enhancement is described by:

$$o_s = o_v\left(k_{sc} + k_{ss}\left(1 - abs\left(\nabla_{fn} \cdot V\right)\right)^{k_{se}}\right)$$

where k_{sc} controls the scaling of non-silhouette regions, k_{ss} controls the amount of silhouette enhancement, and k_{se} controls the sharpness of the silhouette curve.

Figure 4 shows the result of both boundary and silhouette enhancement in the medical volume. The fine honeycomb structure of the liver interior is clearly apparent, as well as additional internal structure of the kidneys. Parameter values used in Figure 4 are $k_{gc} = 0.8$, $k_{gs} = 5.0$, $k_{ge} = 1.0$; $k_{sc} = 0.9$, $k_{ss} = 50$, $k_{se} = 0.25$.

Decreasing the opacity of volume features oriented toward the viewer emphasizes feature orientation, and in the extreme cases, can create sketches of the volume, as illustrated in Figure 5. Figure 5 shows a black and white sketch of the medical dataset by using a white sketch color and making non volumetric silhouettes transparent. To get appropriate shadowing of the sketch lines, the shadows are calculated based on the original volume opacity. Using a black silhouette color can also be effective for outlining volume data.

Orientation information can also be used effectively to change feature color. For instance, in medical illustration the portions of anatomical structures oriented toward the viewer are desaturated and structures oriented away from the view are darkened and saturated [Clark99]. We simulate these effects by allowing the volumetric gradient orientation to the viewer to modify the color, saturation, value, and transparency of the given volume sample. The use of the HSV color space allows the system to easily utilize the intuitive color modification techniques of painters and illustrators. Figure 10 shows oriented changes in the saturation and value of the medical volume. In this figure, the color value (V) is decreased as the angle between the gradient and the viewer increases, simulating more traditional illustration techniques of oriented fading.

5 Depth and Orientation Cues

Few of the usual depth cues are present in traditional rendering of translucent volumes. Obscuration cues are largely missing since there are no opaque objects to show a clear depth ordering. Perspective cues from converging lines and texture compression are also lacking, since few volume models contain straight lines or uniform textures. The dearth of clear depth cues makes

understanding spatial relationships of features in the volume difficult. One common approach to this difficulty is the use of hard transfer functions, those with rapidly increasing opacity at particular value ranges of interest. While this may increase depth cues by creating the appearance of surfaces within the volume, it does so by hiding all information in some regions of the volume, sacrificing a key advantage of volume rendering.

Similarly, information about the orientation of features within the volume is also largely missing. Many volume rendering systems use very simple illumination models and often do not include the effect of shadows, particularly volume self-shadowing to improve performance, even though many volume shadowing algorithms have been developed [Ebert90, Kajiya84]. Accurate volumetric shadowing often produces subtle effects which do not provide strong three-dimensional depth cues. As a result, the shape of individual structures within even illuminated volumes is difficult to perceive.

We have developed several techniques for the enhancement of depth and orientation cues in volume models, inspired by shading concepts in art and technical illustration.

5.1 Distance color blending

Intensity depth-cuing is a well known technique for enhancing the perception of depth in a scene [Foley96]. This technique dims the color of objects far from the viewer, creating an effect similar to viewing the scene through haze. We have adapted this technique for volume rendering, dimming volume sample colors as they recede from the viewer. In addition, we have augmented the standard intensity depth-cuing with a subtle color modulation. This color modulation increases the amount of blue in the colors of more distant volume samples, simulating techniques used for centuries by painters, such as aerial perspective [daVinci1506, Beirstadt1881]. This technique exploits the tendency of cool colors (such as blue) to recede visually while warm colors (such as red) advance.

Depth-cued colors start as the voxel color at the front of the volume, decreasing in intensity and moving toward the background color as depth into the volume increases. The progression of depth-cuing need not be linear; we use an exponential function to control the application of depth-cuing. The distance color blending process can be described by:

$$c_d = \left(1 - k_{ds} d_v^{\,k_{de}}\right) c_v + k_{ds} d_v^{\,k_{de}} c_b$$

where k_{ds} controls the size of the color blending effect, k_{de} controls the rate of application of color blending, d_v is the fraction of distance through the volume, and c_b is a defined background color. When c_b is a shade of grey ($c_b = (a, a, a)$ for some value of a), only standard intensity depth-cuing is performed. Using a background color that is a shade of blue ($c_b = (a, b, c)$ for $c > a$, b), introduces a cool shift in distant regions. Other color modulation effects are clearly possible, but make less sense perceptually.

Figure 6 shows the effect of distance color blending. The ribs behind the lungs fade into the distance and the region around the kidneys seems to recede slightly. Color blending parameters used in Figure 6 are $c_b = (0, 0, 0.15)$, $k_{ds} = 1.0$, $k_{de} = 0.5$.

5.2 Feature halos

Illustrators sometimes use null halos around foreground features to reinforce the perception of depth relationships within a scene. The effect is to leave the areas just outside surfaces empty, even if an accurate depiction would show a background object in that place. Interrante [Interrante98] used a similar idea to show depth relationships in 3D flow data using Line Integral Convolution (LIC). She created a second LIC volume with a larger element size, using this second volume to impede the view. Special care was required to keep objects from being obscured by their own halos. The resulting halos achieved the desired effect, but the method depended on having flow data suitable for processing with LIC.

We introduce a more general method for creating halo effects during the illumination process using the local spatial properties of the volume. Halos are created primarily in planes orthogonal to the view vector by making regions just outside features darker and more opaque, obscuring background elements which would otherwise be visible. The strongest halos are created in empty regions just outside (in the plane perpendicular to the view direction) of a strong feature.

The halo effect at a voxel is computed from the distance weighted sum of haloing influences in a specified neighborhood. In order to restrict halos to less interesting regions, summed influences are weighted by the complement of the voxel's gradient. The size of the halo effect is given by:

$$h_i = \left(\sum_{n}^{neighbors} \frac{h_n}{\|P_i - P_n\|^2} \right) \left(1 - \|\nabla_f(P_i)\|\right)$$

where h_n is the maximum potential halo contribution of a neighbor. The haloing influence of a neighbor is inversely related to its distance and the tendency of a location to be a halo is inversely related to its gradient magnitude.

The maximum potential halo contribution of each neighbor is proportional to the product of the alignment of the neighbor's gradient with the direction to the voxel under consideration (calculated from the dot product between them) and the degree to which the neighbor's gradient is aligned perpendicular to the view direction (also calculated as a dot product). The halo potential (h_n) is given by:

$$h_n = \left(\nabla_{fn}(P_n) \cdot \left(\frac{(P_i - P_n)}{\|P_i - P_n\|} \right) \right)^{k_{hpe}} \left(1 - \nabla_{fn}(P_n) \cdot V\right)^{k_{hse}}$$

where k_{hpe} controls how directly the neighbor's gradient must be oriented toward the current location, and k_{hse} controls how tightly halos are kept in the plane orthogonal to the view direction. The most strong halo effects will come from neighbors that are displaced from the volume sample of interest in a direction orthogonal to the view direction and that have a large gradient in the direction of this sample.

Once the size of the halo effect has been determined, parameters control the visual appearance of halo regions. The most common adjustment to the halo region is to decrease the brightness by a scalar times the halo effect and increase the opacity by another scalar times the halo effect. This method produces effects similar to those of Interrante, but can be applied to any type of data or model during the illumination process. Since the halos generated are inherently view dependent, no special processing must be done to keep features from casting a halo on themselves.

Figure 6 shows the effectiveness of adding halos to the medical volume. Structures in the foreground, such as the liver and kidneys, stand out more clearly. Halo parameters used in Figure 6 are $k_{hpe} = 1.0$ and $k_{hse} = 2.0$.

5.3 Tone shading

Another illustrative technique used by painters is to modify the tone of an object based on the orientation of that object relative to

the light. This technique can be used to give surfaces facing the light a warm cast while surfaces not facing the light get a cool cast, giving effects suggestive of illumination by a warm light source, such as sunlight. Gooch et al. proposed an illumination model based on this technique [Gooch98], defining a parameterized model for effects from pure tone shading to pure illuminated object color. The parameters define a warm color by combining yellow and the scaled fully illuminated object color. Similarly, a cool color combines blue and the scaled ambient illuminated object color. The final surface color is formed by interpolation between the warm and cool color based on the signed dot product between the surface normal and light vector. The model assumes a single light source, generally located above the scene.

We implemented an illumination model similar to Gooch tone shading for use with volume models. As with Gooch tone shading, the tone contribution is formed by interpolation between the warm and cool colors based on the signed dot product between the volume sample gradient and the light vector. Unlike Gooch tone shading, the illuminated object contribution is calculated using only the positive dot product, becoming zero at orientations orthogonal to the light vector. This more closely matches familiar diffuse illumination models.

The color at a voxel is a weighted sum of the illuminated gaseous color (including any traditional transfer function calculations) and the total tone and directed shading from all directed light sources. The new tone illumination model is given by:

$$c = k_{ta} I_G + \sum_{i}^{N_L} \left(I_t + k_{td} I_o \right)$$

where k_{ta} controls the amount of gaseous illumination (I_G) included, N_L is the number of lights, k_{td} controls the amount of directed illumination included, I_t is the tone contribution to volume sample color, and I_o is the illuminated object color contribution. Although this model allows for multiple light sources, more than a few is likely to result in confusing images, since we are not used to interpreting complex illumination coming from many lights.

The tone contribution from a single light source is interpolated from the warm and cool colors, depending on the angle between the light vector and the sample gradient. It is given by:

$$I_t = \left(\left(1.0 + \nabla_{fn} \cdot L\right)/2 \right) c_w + \left(1 - \left(1.0 + \nabla_{fn} \cdot L\right)/2 \right) c_c$$

where L is the unit vector in the direction of the light and

$c_w = (k_{ty}, k_{ty}, 0)$, $c_c = (0, 0, k_{tb})$

describe the warm and cool tone colors. Samples oriented toward the light become more like the warm color while samples oriented away from the light become more like the cool color.

The directed illumination component is related to the angle between the voxel gradient and the light direction, for angles up to 90 degrees. It is given by:

$$I_o = \begin{cases} k_{td} I_i \left(\nabla_{fn} \cdot L \right) : \nabla_{fn} \cdot L > 0 \\ 0 : \nabla_{fn} \cdot L \le 0 \end{cases}$$

where k_{td} controls how much directed illumination is added.

Figure 7 shows modified tone shading applied to the uncolored medical volume. The small structure of the liver shows clearly, as does the larger structures of the kidney. The bulges of intestine at the lower right are much more clearly rounded 3D shapes than with just boundary and silhouette enhancement (Figure 4). Figure 8 shows tone shading applied together with colors from a transfer function. The tone effects are subtler, but still improve shape perception. The basic tissue colors are preserved, but the banded structure of the aorta is more apparent than in a simple illuminated and color-mapped image (Figure 2). Tone shading parameters used in Figures 7 and 8 are $k_{ty} = 0.3$, $k_{tb} = 0.3$, $k_{ta} = 1.0$, $k_{td} = 0.6$.

6 Application Examples

We have also applied the techniques in the previous sections to several other scientific data sets. Figures 10 and 11 are volume rendered images from a 256x256x64 MRI dataset of a tomato from Lawrence Berkeley National Laboratories. Figure 10 is a normal gas-based volume rendering of the tomato where a few of the internal structures are visible. Figure 11 has our volume illustration gradient and silhouette enhancements applied, resulting in a much more revealing image showing the internal structures within the tomato. Parameters used in Figure 11 are $k_{gc} = 0.5$, $k_{gs} = 2.5$, $k_{ge} = 3.0$; $k_{sc} = 0.4$, $k_{ss} = 500$, $k_{se} = 0.3$.

Figure 12 shows a 512x512x128 element flow data set from the time series simulation of unsteady flow emanating from a 2D rectangular slot jet. The 2D jet source is located at the left of the image and the flow is to the right. Flow researchers notice that both Figures 12 and 13 resemble Schlieren photographs that are traditionally used to analyze flow. Figure 13 shows the effectiveness of boundary enhancement, silhouette enhancement, and tone shading on this data set. The overall flow structure, vortex shedding, and helical structure are much easier to perceive in Figure 13 than in Figure 12.

Figures 14 and 15 are volume renderings of a 64x64x64 high-potential iron protein data set. Figure 14 is a traditional gas-based rendering of the data. Figure 15 has our tone shading volume illustration techniques applied, with parameters $k_{ty} = 0.15$, $k_{tb} = 0.15$, $k_{ta} = 1.0$, $k_{td} = 0.6$. The relationship of structure features and the three-dimensional location of the features is much clearer with the tone-based shading enhancements applied.

7 Conclusions

We have introduced the concept of volume illustration, combining the strengths of direct volume rendering with the expressive power of non-photorealistic rendering techniques. Volume illustration provides a powerful unified framework for producing a wide range of illustration styles using local and global properties of the volume model to control opacity accumulation and illumination. Volume illustration techniques enhance the perception of structure, shape, orientation, and depth relationships in a volume model. Comparing standard volume rendering (Figures 2, 10, 12, 14) with volume illustration images (Figures 3, 4, 5, 6, 7, 8, 9, 11, 13, 15) clearly shows the power of employing volumetric illustration techniques to enhance 3D depth perception and volumetric feature understanding.

8 Future Directions

We plan on extending our collection of NPR techniques and exploring suitability of these volume illustration techniques for data exploration and diagnosis.

9 Acknowledgements

We would like to thank researchers at the Mississippi State University NSF Computational Field Simulation Engineering Research Center and the Armed Forces Institute of Pathology for help in evaluating the effectiveness of these technique and

guiding our research. We would also like to thank Dr. Elliott Fishman of Johns Hopkins Medical Institutions for the abdominal CT dataset. The iron protein dataset came from the vtk website (www.kitware.com/vtk.html). Christopher Morris generated some to the pictures included in this paper. This work supported in part by the National Science Foundation under Grants ACIR 9996043 and ACIR 9978032.

References

[Bierstadt1881] Albert Bierstadt. "Near Salt Lake City, Utah," Museum of Art, Brigham Young University, 1881.

[Clark99] John O.E. Clark. *A Visual Guide to the Human Body, Barnes and Noble Books, 1999.*

[daVinci1506] Leonardo daVinci, "The Virgin of the Rocks," National Gallery, London, 1503-1506.

[Drebin88] Robert A. Drebin and Loren Carpenter and Pat Hanrahan. Volume Rendering, *Computer Graphics (Proceedings of SIGGRAPH 88),* 22(4), pp. 65-74 (August 1988, Atlanta, Georgia). Edited by John Dill.

[Ebert90] David S. Ebert and Richard E. Parent. Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques, *Computer Graphics (Proceedings of SIGGRAPH 90),* 24 (4), pp. 357-366 (August 1990, Dallas, Texas). Edited by Forest Baskett. ISBN 0-201-50933-4.

[Fang98] Shiaofen Fang and Tom Biddlecome and Mihran Tuceryan. Image-Based Transfer Function Design for Data Exploration in Volume Visualization, *IEEE Visualization '98,* pp. 319-326 (October 1998). IEEE. Edited by David Ebert and Hans Hagen and Holly Rushmeier. ISBN 0-8186-9176-X.

[Foley96] James Foley, Andries van Dam, Steven Feiner, and John Hughes, *Computer Graphics: Principles and Practice, Second Edition in C,* Addison Wesley 1996.

[Fujishiro99] Issei Fujishiro and Taeko Azuma and Yuriko Takeshima. Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis, *IEEE Visualization '99,* pp. 467-470 (October 1999, San Francisco, California). IEEE. Edited by David Ebert and Markus Gross and Bernd Hamann. ISBN 0-7803-5897-X.

[Gooch98] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-photorealistic Lighting Model for Automatic Technical Illustration. In *Proceedings of SIGGRAPH '98* (Orlando, FL, July 1998), Computer Graphics Proceedings, Annual Conference Series, pp. 447-452, ACM SIGGRAPH, ACM Press, July 1998.

[Gooch99] Bruce Gooch and Peter-Pike J. Sloan and Amy Gooch and Peter Shirley and Rich Riesenfeld. Interactive Technical Illustration, *1999 ACM Symposium on Interactive 3D Graphics,* pp. 31-38 (April 1999). ACM SIGGRAPH. Edited by Jessica Hodgins and James D. Foley. ISBN 1-58113-082-1 .

[Interrante95] Victoria Interrante, Henry Fuchs, and Stephen Pizer. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines, IEEE Visualization '95, pp. 52-59 (October 1995, Atlanta GA). IEEE. Edited by Gregory Nielson and Deborah Silver. ISBN 0-8186-7187-4.

[Interrante97] Victoria Interrante and Henry Fuchs and Stephen M. Pizer. Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture, *IEEE Transactions on Visualization and Computer Graphics,* 3(2), (April - June 1997). ISSN 1077-2626.

[Interrante98] Victoria Interrante and Chester Grosch. Visualizing 3D Flow, *IEEE Computer Graphics & Applications,* 18(4), pp. 49-53 (July - August 1998). ISSN 0272-1716.

[Kajiya84] James T. Kajiya and Brian P. Von Herzen. Ray Tracing Volume Densities, *Computer Graphics (Proceedings of SIGGRAPH 84),* 18(3), pp. 165-174 (July 1984, Minneapolis, Minnesota). Edited by Hank Christiansen.

[Kindlmann98] Gordon Kindlmann and James Durkin. Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering, In *Proceedings of 1998 IEEE Symposium on Volume Visualization,* pp. 79-86.

[Kirby99] R.M. Kirby, H. Marmanis, and D.H. Laidlaw. Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting, *IEEE Visualization '99,* pp. 333-340 (October 1999, San Francisco, California). IEEE. Edited by David Ebert and Markus Gross and Bernd Hamann. ISBN 0-7803-5897-X.

[Krueger91] Wolfgang Krueger, The Application of transport theory to the visualization of 3D scalar fields, *Computers in Physics,* pp. 397-406, July 1991.

[Laidlaw98] David H. Laidlaw and Eric T. Ahrens and David Kremers and Matthew J. Avalos and Russell E. Jacobs and Carol Readhead. Visualizing Diffusion Tensor Images of the Mouse Spinal Cord, *IEEE Visualization '98,* pp. 127-134 (October 1998). IEEE. Edited by David Ebert and Hans Hagen and Holly Rushmeier. ISBN 0-8186-9176-X.

[Levoy90] Marc Levoy. Efficient Ray Tracing of Volume Data, *ACM Transactions on Graphics,* 9 (3), pp. 245-261 (July 1990). ISSN 0730-0301.

[Max95] Nelson Max. Optical models for direct volume rendering, *IEEE Transactions on Visualization and Computer Graphics,* 1 (2), pp. 99-108 (June 1995). ISSN 1077-2626.

[Nishita87] Tomoyuki Nishita and Yasuhiro Miyawaki and Eihachiro Nakamae. A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources, *Computer Graphics (Proceedings of SIGGRAPH 87),* 21 (4), pp. 303-310 (July 1987, Anaheim, California). Edited by Maureen C. Stone.

[Nishita98] Tomoyuki Nishita. Light Scattering Models for the Realistic Rendering of Natural Scenes, *Eurographics Rendering Workshop 1998,* pp. 1-10 (June 1998, Vienna, Austria). Eurographics. Edited by George Drettakis and Nelson Max. ISBN3-211-83213-0.

[Rheingans96] Penny Rheingans. Opacity-modulating Triangular Textures for Irregular Surfaces, Proceedings of IEEE Visualization '96, pp. 219-225 (October 1996, San Francisco CA). IEEE. Edited by Roni Yagel and Gregory Nielson. ISBN 0-89791-864-9.

[Saito90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3-D Shapes, *Computer Graphics (Proceedings of SIGGRAPH 90),* 24 (4), pp. 197-206 (August 1990, Dallas, Texas).

[Saito94] Takafumi Saito. Real-time Previewing for Volume Visualization. In *Proceedings of 1994 IEEE Symposium on Volume Visualization,* pp. 99-106.

[Salisbury94] Michael P. Salisbury and Sean E. Anderson and Ronen Barzel and David H. Salesin. Interactive Pen-And-Ink Illustration, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series,* pp. 101-108 (July 1994, Orlando, Florida). ACM Press. Edited by Andrew Glassner. ISBN 0-89791-667-0.

[Salisbury97] Michael P. Salisbury and Michael T. Wong and John F. Hughes and David H. Salesin. Orientable Textures for Image-Based Pen-and-Ink Illustration, *Proceedings of SIGGRAPH 97,* Computer Graphics Proceedings, Annual

Conference Series, pp. 401-406 (August 1997, Los Angeles, California). Addison Wesley. Edited by Turner Whitted. ISBN 0-89791-896-7.

[Treavett] S.M.F. Treavett and M. Chen. Pen-and-Ink Rendering in Volume Visualisation, *Proceedings of IEEE Visualization 20000*, October 2000, ACM SIGGRAPH Press.

[Williams92] Peter L. Williams and Nelson Max. A Volume Density Optical Model, *1992 Workshop on Volume Visualization*, pp. 61-68 (1992). ACM.

[Winkenbach94] Georges Winkenbach and David H. Salesin. Computer-Generated Pen-And-Ink Illustration, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series*, pp. 91-100 (July 1994, Orlando, Florida). ACM Press. Edited by Andrew Glassner. ISBN 0-89791-667-0.

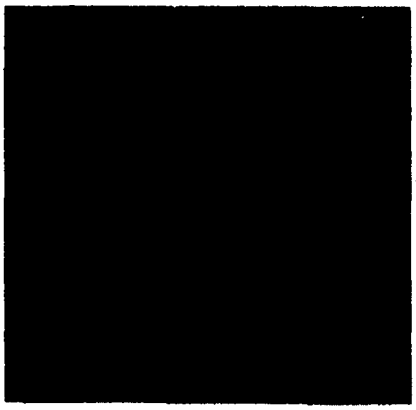

Figure 1. Gaseous illumination of medical CT volume. Voxels are a constant color.

Figure 5. Volumetric sketch lines on CT volume. Lines are all white.

Figure 12. Atmospheric volume rendering of square jet. No illustration enhancements.

Figure 2. Gaseous illumination of color-mapped CT volume.

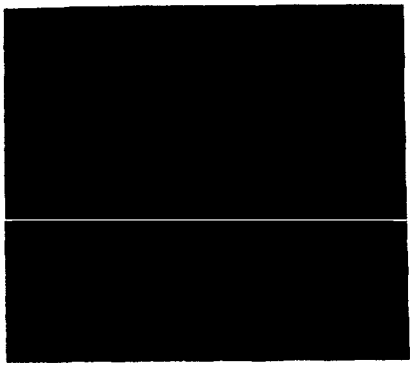

Figure 3. Color-mapped gaseous illumination with boundary enhancement.

Figure 4. Silhouette and boundary enhancement of CT volume.

Figure 6. Distance color blending and halos around features of CT volume.

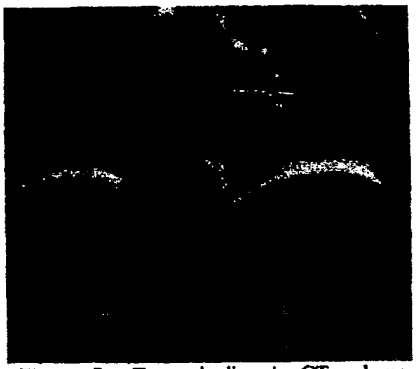

Figure 7. Tone shading in CT volume. Surfaces toward light receive warm color.

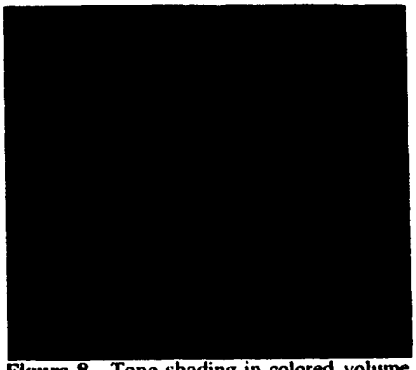

Figure 8. Tone shading in colored volume. Surfaces toward light receive warm color.

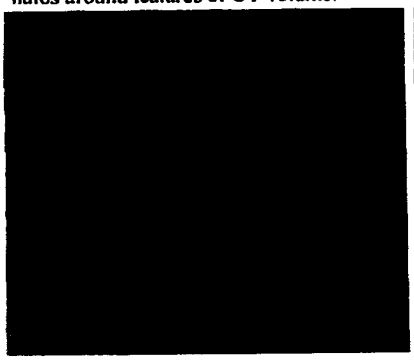

Figure 9. Orientation fading. Surfaces toward viewer are desaturated.

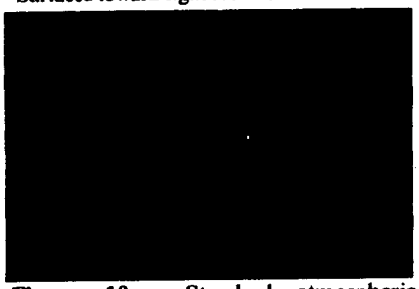

Figure 10. Standard atmospheric volume rendering of tomato.

Figure 11. Boundary and silhouette enhanced tomato.

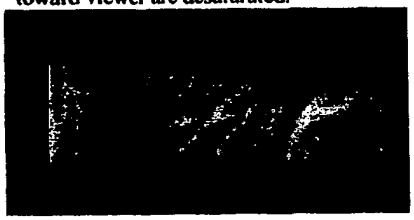

Figure 13. Square jet with boundary and silhouette enhancement, and tone shading.

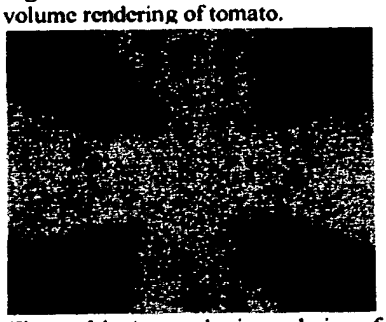

Figure 14. Atmospheric rendering of iron protein.

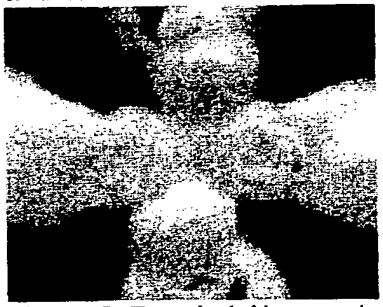

Figure 15. Tone shaded iron protein.